

AMENDMENTS IN THE CLAIMS

1. (currently amended) In a[[n]] multiprocessor data processing system (MP) ~~configured according to IA-32 architecture~~, a method for dynamically providing spare processor resources when an operating processor fails, said method comprising:

holding-off a spare processor during an initial POST (power on self test), wherein said spare processor is available within said MP along with at least two operating processors, and said spare processor is not allocated any processing load by the operating system (OS) following the POST; and

when one of the operating processors is determined to be failing, dynamically activating said spare processor to replace the failing operating processor, wherein the processing load of the failing processor is automatically sent to the spare processor for processing.

2. (currently amended) The method of Claim 1, wherein:

said MP includes a basic input/output system (BIOS) and a processor register linked to said BIOS, which indicates which processors among said operating processors and said spare processors are currently available to said OS for allocating load; and[[,]]

~~wherein~~ said holding off of the spare processor comprises:

setting a bit within said processor register to a first value corresponding to assigning each of said operating processors to an active state during said initial POST, wherein said active state indicates to said OS that the corresponding operating processor is available for allocating load; and

setting a next bit within the processor register to a second value corresponding to assigning said spare processor to an inactive state.

3. (currently amended) The method of Claim 2, wherein said dynamically activating step comprises:
 - re-setting a bit corresponding to the failing processor to an inactive state; and
 - setting said next bit corresponding to said spare processor to the first value indicating an active state.
4. (original) The method of Claim 1, wherein said holding off step comprises:
 - placing said spare processor in a low-power, standby state during system boot utilizing advanced configuration and power interface (ACPI) of the BIOS; and
 - subsequently handing off control of said MP from said BIOS to the OS.
5. (original) The method of Claim 1, wherein said holding off step comprises:
 - placing said spare processor in an S3 state during system boot utilizing advanced configuration and power interface (ACPI) of the BIOS; and
 - subsequently handing off control of said MP from said BIOS to the OS.
6. (currently amended) The method of Claim 1, wherein said holding off step further comprises:
 - bringing said spare processor online during BIOS boot; and
 - [[then]] placing the spare processor into a spin lock.
7. (original) The method of Claim 1, further comprising generating a message signaling said processor failure to a system administrator.
8. (currently amended) The method of Claim 1, wherein said MP is configured according to IA-32 architecture and said dynamically activating step comprises completing a driver-OS-SMI handshake operation.

9. (currently amended) The method of Claim 1, wherein said MP is configured according to IA-32 architecture and said method further comprises comprising:

activating a system management interrupt (SMI) when said failing processor is determined to be failing, wherein said SMI messages said OS to hold off allocation of said processing load to said failing processor.

10. (original) The method of Claim 9, further comprising:

saving a system state of each processor to memory;
activating an SMI handler to temporarily take control of said system from said OS while said spare processor is being activated;
swapping out the failing processor for the spare processor;
updating system configuration parameters, including bus controller, memory and OS parameters, to reflect switch over to said spare processor from said failing processor; and
restoring control to said OS once said spare processor has been activated.

11. (original) The method of Claim 9, further comprising enabling OS updates to an active processor lists, wherein said OS begins allocating threads previously running on said failed processor to said spare processor.

12. (currently amended) A multiprocessor data processing system (MP) configured according to IA-32 architecture, comprising:

a plurality of processors having at least two main operating processors and a spare processor;

a memory coupled to said plurality of processors;

an operating system (OS) that controls work allocation to each of said plurality of processors;

a basic input output system (BIOS) that supports system management interrupt (SMI);

means for holding-off the spare processor during an initial POST (power on self test), wherein said spare processor is not allocated any processing load by the operating system (OS) following a power-on self test (POST); and

means, when one of the operating processors is determined to be failing, for dynamically activating said spare processor to replace the failing operating processor, wherein the processing load of the failing processor is automatically sent to the spare processor for processing.

13. (currently amended) The MP of Claim 12, further comprising:

a processor register linked to said BIOS, which indicates which processors among said operating processors and said spare processors are currently available to said OS for allocating load; [[and]]

means for setting a bit within said processor register to a first value corresponding to assigning each of said operating processors to an active state during said initial POST, wherein said active state indicates to said OS that the corresponding operating processor is available for allocating processing load[[,]]; and

means for setting a next bit within the processor register to a second value corresponding to assigning ~~wherein further a bit of said spare processor is set~~ to an inactive state.

14. (currently amended) The MP of Claim 13, wherein said means for dynamically activating comprises:

means for re-setting a bit corresponding to the failing processor to an inactive state; and

means for setting the next bit corresponding to said spare processor to the first value indicating an active state.

15. (original) The MP of Claim 12, wherein said means for holding off comprises:

means for placing said spare processor in a low-power, standby state during system boot utilizing advanced configuration and power interface (ACPI) of the BIOS; and

means for subsequently handing off control of said MP from said BIOS to the OS.

16. (original) The MP of Claim 12, wherein said means for holding off comprises:

means for placing said spare processor in an S3 state during system boot utilizing advanced configuration and power interface (ACPI) of the BIOS; and

means for subsequently handing off control of said MP from said BIOS to the OS.

17. (original) The MP of Claim 12, wherein said means for holding off further comprises BIOS means for bringing said spare processor online and then placing the spare processor into a spin lock.

18. (currently amended) The MP of Claim ~~[[11]]~~12, further comprising generating a message signaling said processor failure to a system administrator.

19. (currently amended) The MP of Claim 12, wherein said MP is configured according to IA-32 architecture and said means for dynamically activating comprises means for completing a driver-OS-SMI handshake operation.

20. (currently amended) The MP of Claim 12, wherein said MP is configured according to IA-32 architecture and further ~~comprising~~ comprises:

means for activating a system management interrupt (SMI) when said failing processor is determined to be failing, wherein said SMI messages said OS to hold off allocation of said processing load to said failing processor.

21. (original) The MP of Claim 20, further comprising:

means for saving a system state of each processor to memory;

means for activating an SMI handler to temporarily take control of said system from said OS while said spare processor is being activated;

means for swapping out the failing processor for the spare processor;

means for updating system configuration parameters, including bus controller, memory and OS parameters, to reflect switch over to said spare processor from said failing processor; and

means for restoring control to said OS once said spare processor has been activated.

22. (original) The MP of Claim 21, further comprising means for enabling OS updates to an active processor lists, wherein said OS begins allocating threads previously running on said failed processor to said spare processor.

23. (original) The MP of Claim 12, wherein further:

said plurality of processors are configured with said at least two main processors on a first processing board and said spare processor on a second processing board, wherein said second processing board is a processing board among a plurality of processing boards containing a processor within the MP that is most difficult to access.

RPS920030082US1

-8-